

Vision

Make members ready has the ability to dive into any linux fields with their new skills in scripting and contribute on open source projects

Mission

Study the Essential commands, tools and skills to work on linux as advanced users

- Spread Linux knowledge between faculty members through workshops and events.
- Build an appreciation for open source by encouraging contributions to Linux and analysis of source code.
- Foster a community of skilled, passionate and confident Linux powerusers.

SWOT Analysis

Strengths	Weaknesses
- a lot of old members with passion in many linux fields (Lowlevel, Customize, Devops, CTF)	Half of old members in 4th year and the others between 3th year or graduated
- Many Hard workers and good knowledge in linux basics	
Opportunities	Threats
- New Projects and New Ideas to help others	Time problem due to the exams and freezes
- Creating "Get To Git" event to encourage many students to apply.	
Last year's events (summer training and Salakhana) will attract new members with a solid technical foundation and/or passion.	

Recruitments and Interviews

required topics for intermediet and new

<https://docs.google.com/document/d/IwlOO8kZ-mUnneAINzoZwx2b2lYToYXlF2DchyYpmOQk/edit?tab=t.o>

Members

recruitment, old members will prepare interview questions and brush up their knowledge on the covered concepts.

A week before recruitment, mock interviews will be held between Linux committee members.

Level 1: Beginners

Goal:

- Build appreciation for Linux and open source
- Build strong Linux skills and learn how to apply them to become a more efficient software engineer.
- Familiarize them with source control and Git.
- Build leadership and communication skills by encouraging them to spread their knowledge (supporting, giving sessions, etc).

Criteria:

- Basic knowledge about Linux and open source
- Basic terminal experience and familiarity with commands like (ls, cd, pwd, mv, touch, man) We mean at least the first five or four sessions in playlist of Linux
- Interest and curiosity about Linux.

Level 2: Intermittent

Goal:

- Strengthen fundamentals and introduce more advanced topics early on.
- Build mentorship skills.

Criteria

- Roughly the same criteria for summer training evaluation, OR Using Linux as a daily driver

Level 3: old

Goal:

- Further build their technical skills by teaching them more advanced content.

- Build leadership and communication skills by encouraging them to spread their knowledge (mentoring new members, leading events, giving sessions, etc).
- Help them convert their passion and skills to either:
 - A career path
 - Practical skills to elevate their productivity
- Build projects

Expected new member count

~20 members.

~5 intermediates

~15 beginners

Events

FOR OSC

Get to Git OSC

| WorkShop for OSC to learn how to use git and gihtub

Get to Linux OSC

- Expected date : Beginning of the first semester.
- Sessions : 2 sessions, each 3 hours, with task to work on linux VM
 - Introduction to Linux interface / installing apps / working with GUI environment [Genome , KDE] distros
 - introduction to Linux essential commands [navigate between folders, install/uninstall packages, list and change permissions , edit and write into vim/nvim]

FOR Linux Committee

Competitive Linux/Git Commands [For Linux Members]

- Hackathon will be after mid year final on Linux commands and git commands based on qo tool with old questions levels
- old will write during the first semester will contribute on linux-committee-Bank

For Computer Science students

Get to Git

- Expected date : Beginning of the second semester
- Target : First Year Student and any person want to learn how to treat with git
- Sessions : 2 sessions, each 2 hours
 - Introduction to Git
 - Working with others using Git and Github

LDP

- Expected date : As decided by OSC board.
- Spread knowledge about linux distros
- Create VMs and live USBs for people to try Linux.

Distro

Distros:

PopOS

Arch

Kali

EndouverOs

TileOs

Salakhana

- Expected date : As decided by OSC board.
- Show them our power of scripts and some of useful learning games .
- Create VMs and live USBs for people to try Linux.
- Show them our learning games/tool [qo as example]

Linux Summer Training

- Expected date : Summer vacation.
- Sessions : 6 sessions, each 3 hours
- Content : Very similar to '25 summer training
- Evaluation :
 - Tasks
 - CTF at the end

Time Plan

<https://docs.google.com/spreadsheets/d/1Q3HVHcLBm4msCdJRS5IzFIkYFRPO3tPbSrlml2iT6PU/edit?usp=sharing>

Projects

projects by default assigned for old members, but if there's new member has the ability and the skills to contribute on it , So why not m Let's have fun.

qo as tool for competitive/train/eval

qo is powerful tool written in go languages - easy lang to learn and understand - it can be used with some editions to be a more suitable for the hackthon with some edits to be used in bash and git commands and git hard cases - hard cases → make some cases and file that make conflict/problem as virtualization for the real case if you fall into problem - we need to make learn git by the problem and train on problem not just memorize commands

site-quartz as a new edge from BeRoot

BeRoot is a great idea for help people to learn linux as good place contain all required information

But We need to give also readers section collect thee issues and problem faced on linux and how can you solve it and also customize or resources or new topics hard to embedded into the core season

Projects

Package Manager

Shell Checker

raise

Build Regex Engine

git internals

Build your own Docker

boom (CLI Clipboard)

[!NOTE BOOM]

boom manages your text snippets on your command line. You can stash away text like URLs, canned responses, and important notes and then quickly copy them onto your clipboard, ready for pastin

Projects is selected by votes the previous projects is suggests for them but they will give us the idea and make a vote and will take two projects work on `Selected Project 1` and `selected project 2`

Selected Project 1

Selected Project 2

we will have a repository like `Jam Bash` but with some edits

First this repository is for any useful scripts written by members and after the bash session there will be like a table with ideas between easy and hard every script will have rewards according to the difficult of it

Meeting Guidelines For old

- Meetings will be held every two weeks, all members are required to attend.
- The average meeting will be about 3 hours.
- Meetings will include sessions for committee members where certain member(s) called “Moderators of the Meeting” will teach assigned topics to the rest of the committee.
- Every meeting, there will be a discuss about the projects and upcoming sessions for both new and old

Rules

- Mutual and equal respect between all the members, no matter the hierarchy.
- Members must only use Linux in Meetings / events / workshops, whether it is a main OS (maybe VM for somecases), live on USB or in VM (Fullscreen), it will not be allowed otherwise.
- Members must attend meetings, and if they are unable to do so, they must apologize at least a one day before, with an explanation of the reason for the absence.

- Members must come on time for events / workshops /meetings.
- Each session will start at the specified time and latecomers will not be waited.
- Moderators of the session must arrive before the specified time (at least 15 minutes before) to prepare everything they need., if they don't with no serious reason they'll get an alert.
- The instructor of the session must prepare its content at least two days in advance.
- Members must deliver the tasks before the specified date. If they have serious reasons for delay, they must ask for a permission at least three days before.
- Tasks delivery must be done through GitHub using Git through the command line only. It will not be allowed to be sent through any other means.
- Members must react to any message sent on the announcement channel on Discord.
- Checking Discord daily for updates is mandatory.
- Members must attend events with their T-Shirts and IDs.

Rewards

- Easy task → +5 mohsens
- Medium task → +10 mohsens
- Hard task → +20 mohsens
- High-quality task → +5 mohsens
- Task delivered before deadline → +10 mohsens
- Write about Issue and solve it → +10 mohsens
- Push useful Script → +10 mohsens
- Solve an issue in project → +10 mohsens
- Attending on time → +5 mohsens
- Active in meeting → +10 mohsens

Scripts Idea

- Easy script → +5 mohsens
- Medium script → +10 mohsens
- Hard script → +20 mohsens

Badges

- Distro Hopper: Try at least 5 different distros.

- Lazy Admin: Have at least 15 useful Crontabs.
- Unique System: Pick one of your Linux Systems.
- Sensei : Give a Linux session.
- Pro Builder: Install Arch Linux.
- Hell Diver: Install Gentoo.
- Magneto: Install Linux from scratch

Punishments



Alerts follow a 3 strike system. The third alert is equal to a fire.
These apply to old and new members alike. However, in the case of old members, a meeting with the highboard will be held before firing is confirmed.

- Disrespect or misconduct: alert or fire, depending on severity
- Unexcused absence from meeting: alert
- Unexcused lateness to meeting >15 minutes: -10 mohsen
- Using closed-source software in a session: -5 mohsen
- Using closed-source software in an event: -15 mohsen
- Using the GitHub GUI to upload files/tasks (after Git session): -10 mohesn